# End-to-End Containerization of a Flask App using Docker

Author: Aditya Saxena

## 1. Introduction

This report presents a comprehensive step-by-step walkthrough for developing, testing, and containerizing a Python Flask application using Docker. It also includes handling Windows-specific configuration, including WSL integration, and concludes with a comparison of native vs containerized execution.

## 2. Installing Docker Desktop on Windows

To install Docker Desktop:

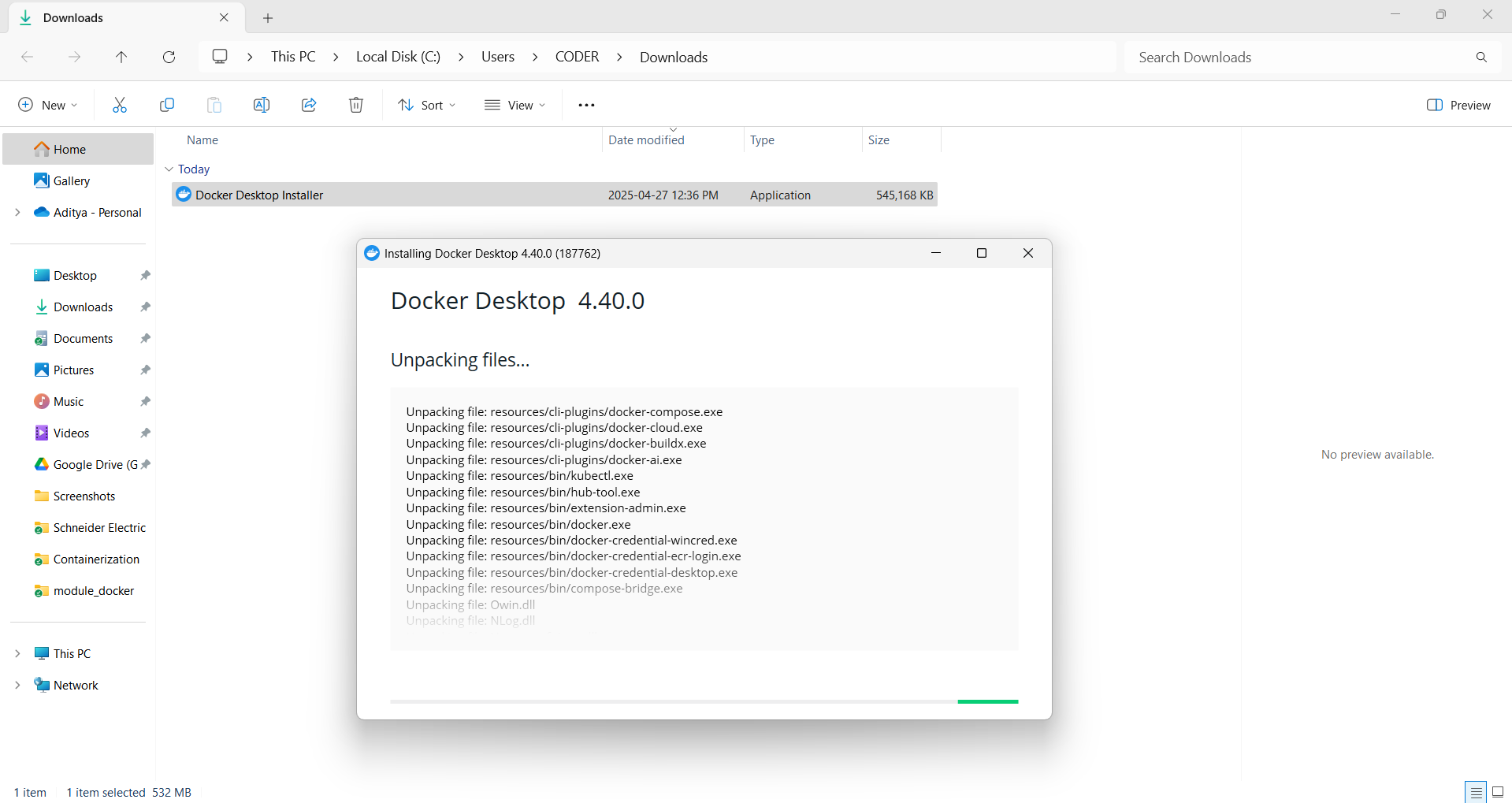
1. Download Docker Desktop from the official site.

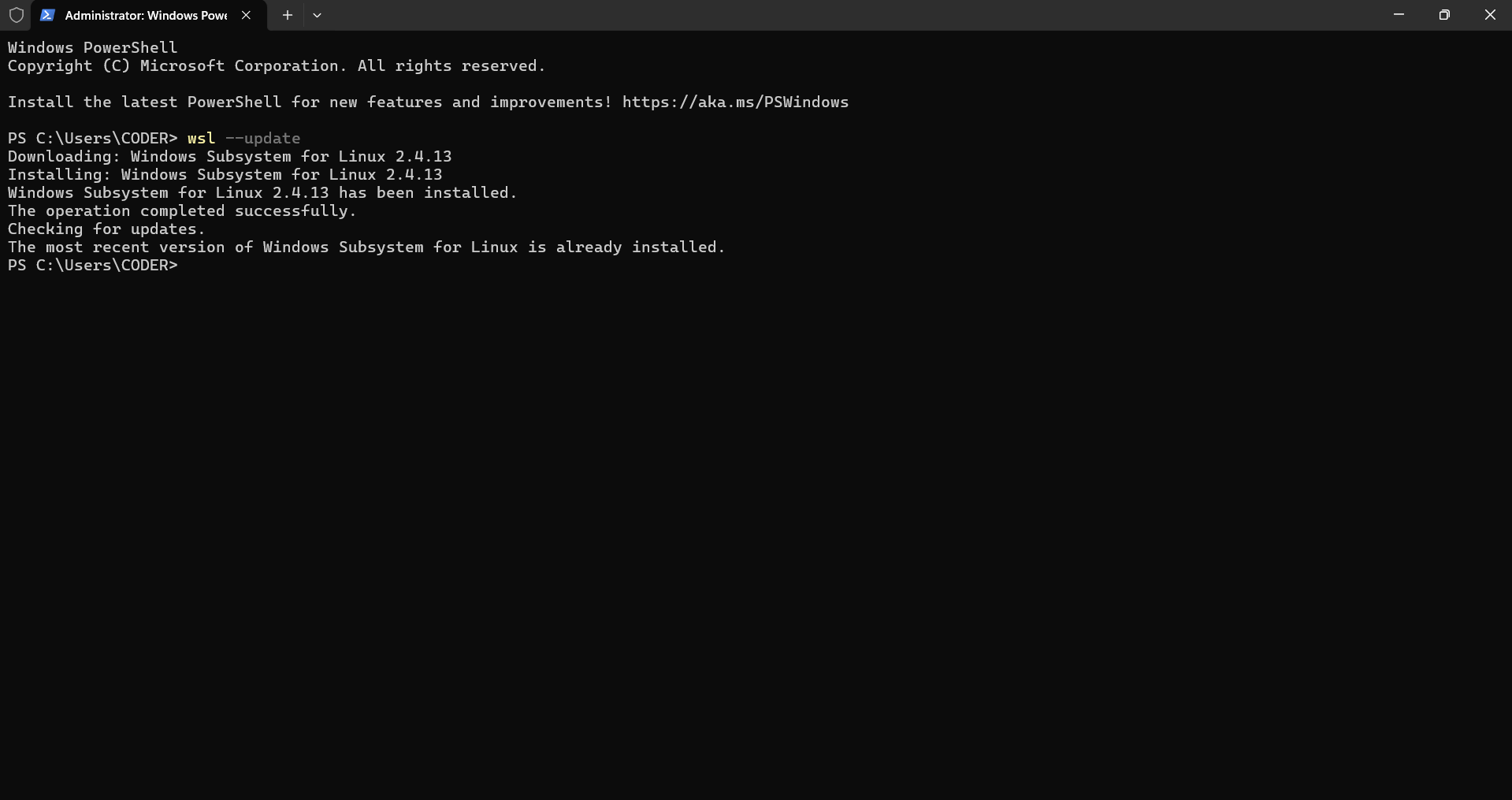
2. Run the installer. Ensure WSL 2 backend is selected.

3. Resolve errors (e.g., WSL update failed) using:

wsl --update

4. Reboot if prompted. Docker Desktop should start without errors.





## 3. Flask App Using venv

A virtual environment was used for local testing:

Commands:

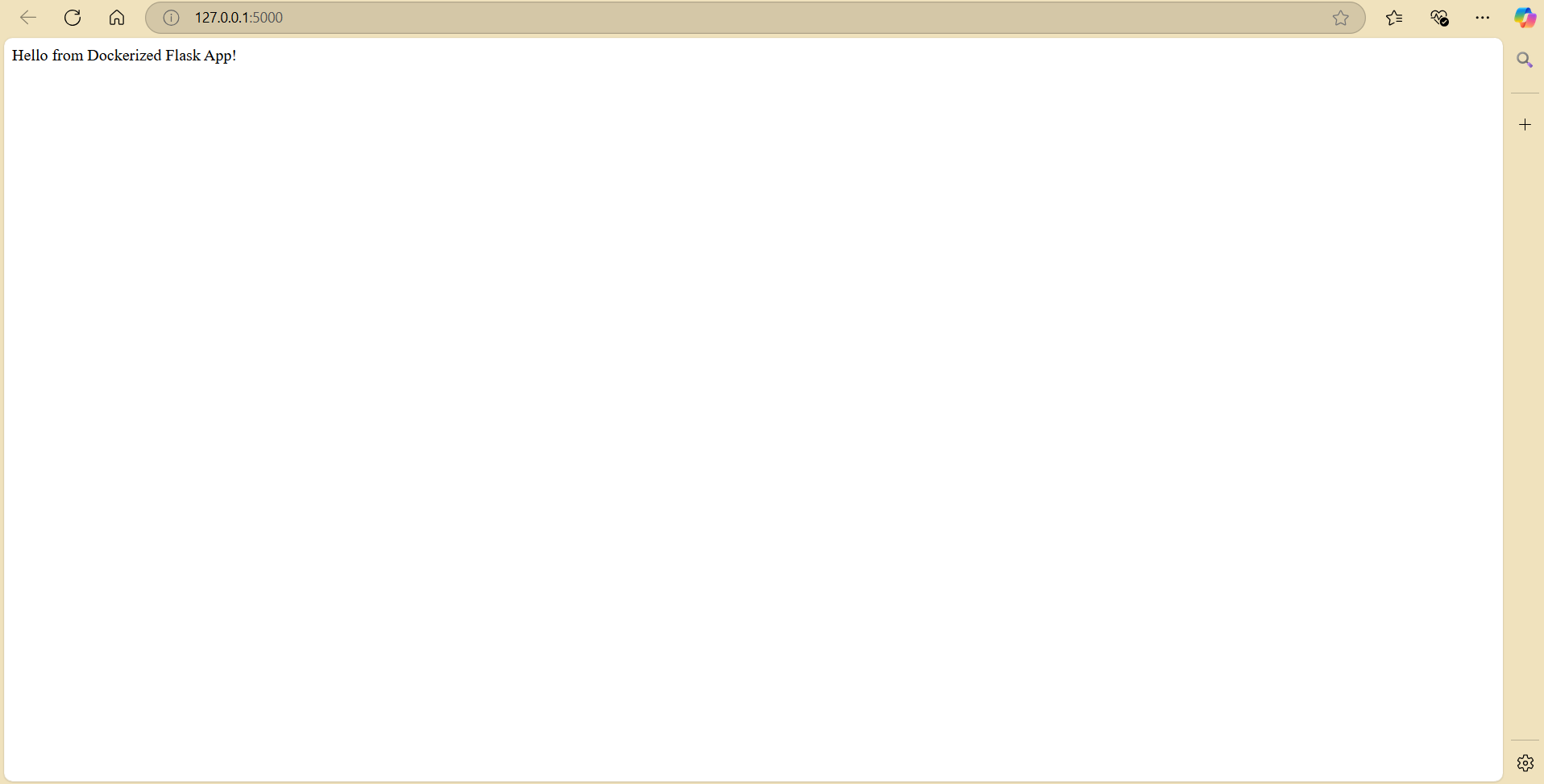
python -m venv venv

.\venv\Scripts\Activate.ps1

pip install -r requirements.txt

python app.py

Screenshot of output:

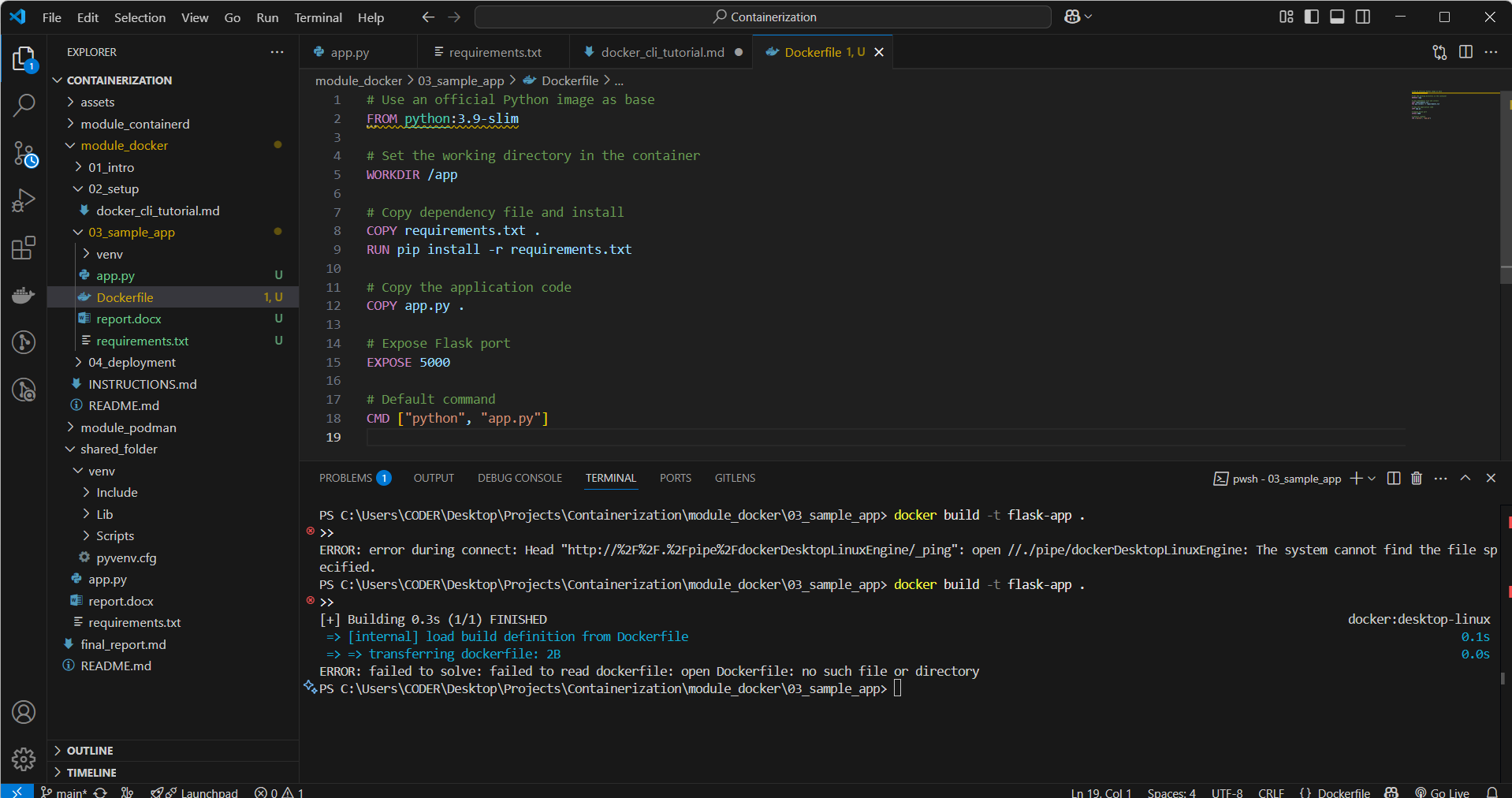


## 4. Dockerfile: Purpose and Structure

The Dockerfile provides Docker with instructions to replicate the app environment inside a container.

Key components:

* - FROM: base image (python:3.9-slim)
* - WORKDIR: set working directory to /app
* - RUN: install dependencies using pip
* - COPY: include source files
* - CMD: default startup command



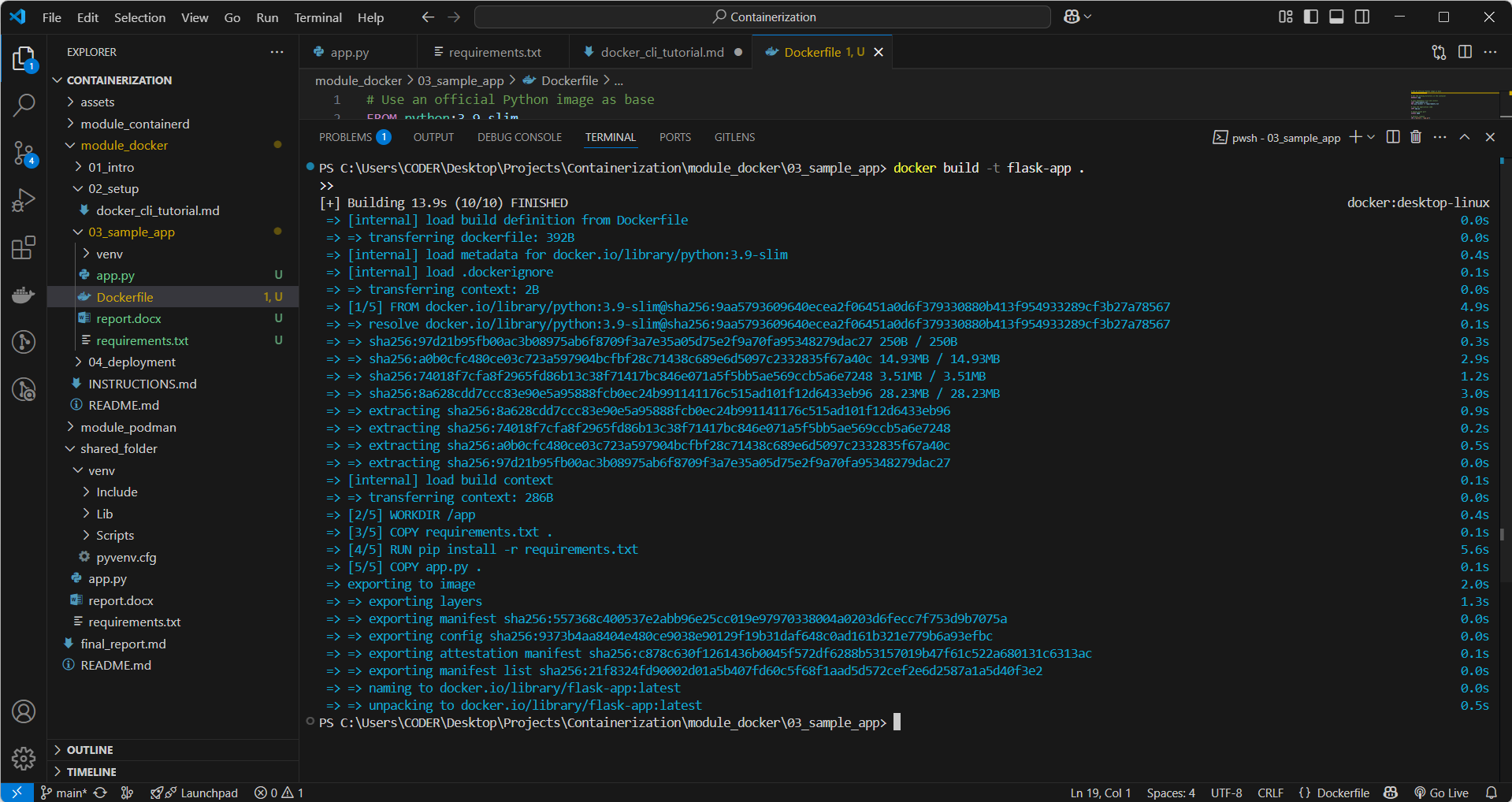
## 5. Building and Running the Docker Container

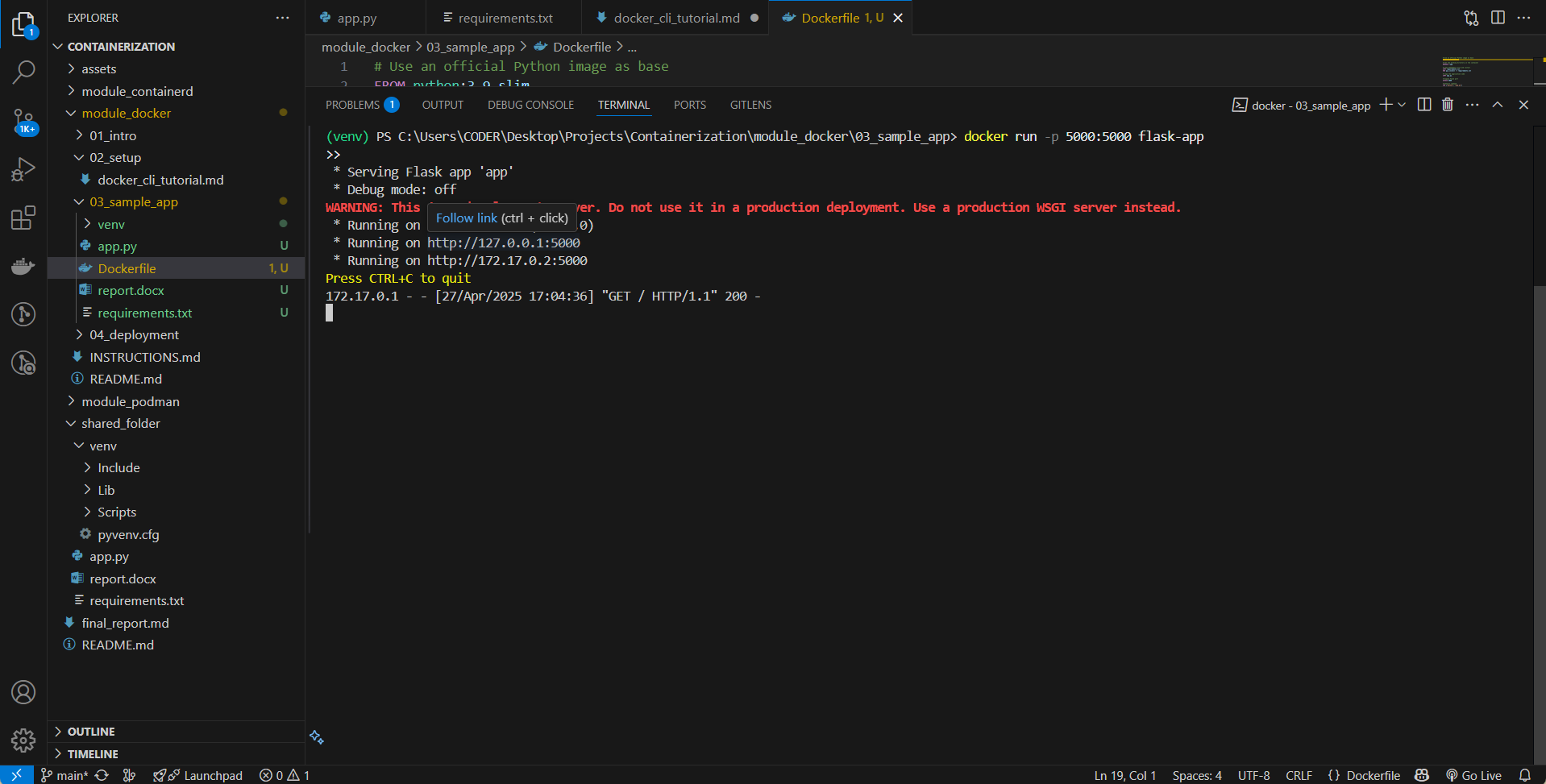
To build the image and run it:

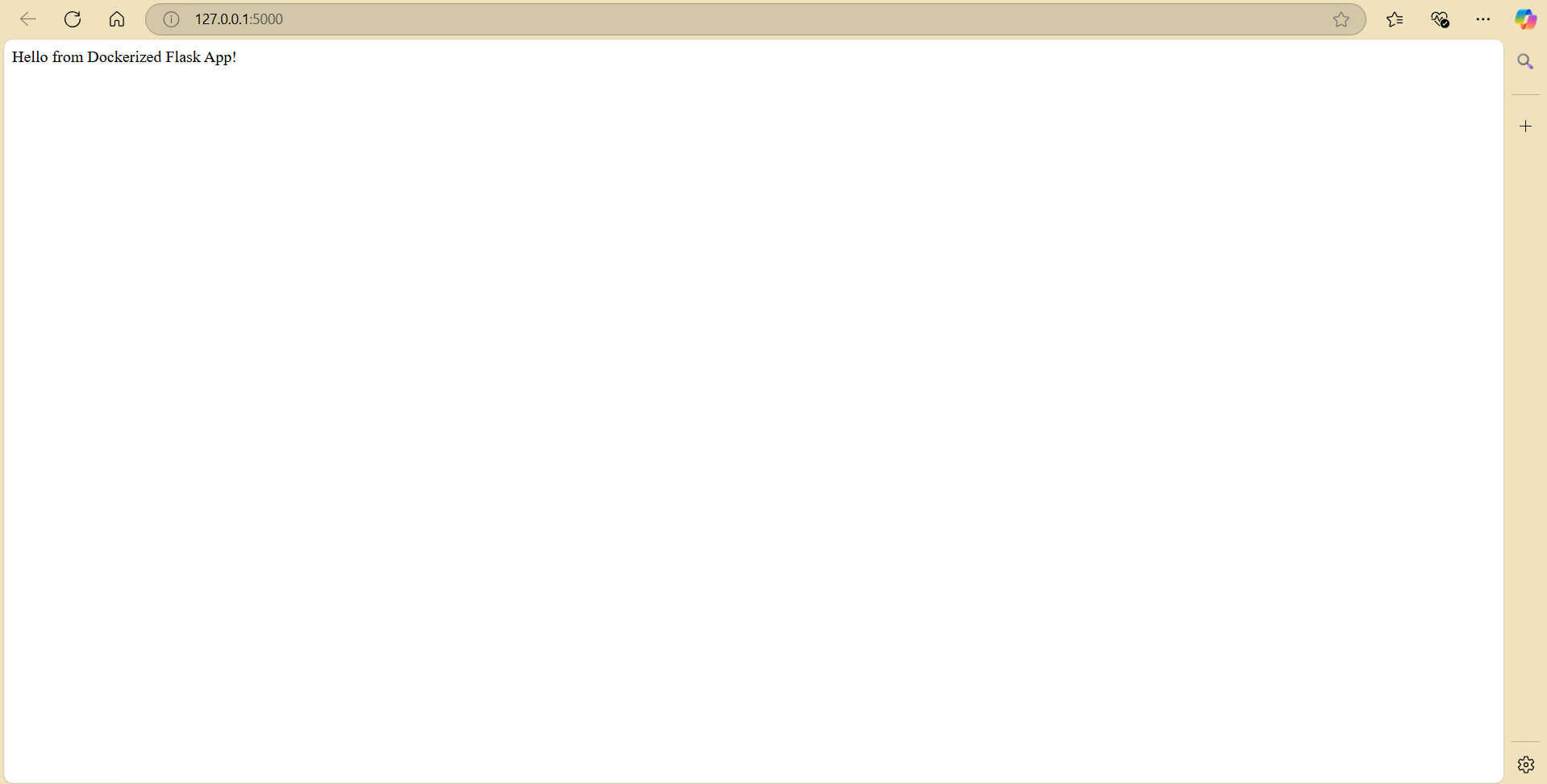
docker build -t flask-app .

docker run -p 5000:5000 flask-app

Images show successful build and run:







## 6. Common Errors and Fixes

- Docker not recognized: Install Docker Desktop and restart shell.

- Docker daemon not running: Start Docker Desktop.

- WSL errors: Run wsl --update and reboot.

## 7. Role of .dockerignore

The .dockerignore file helps avoid copying unnecessary files (e.g., venv/, \_\_pycache\_\_) into the Docker image, reducing image size and avoiding compatibility issues.

## 8. Native vs Containerized Execution

Virtual Environment (venv):

- Tied to local OS and Python version

- Good for development

- Requires manual setup

Docker:

- Fully isolated and OS-agnostic

- More reproducible and scalable

- Suitable for production